山东大学 软件 学院

众智科学与网络化产业 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：201800301249 | 姓名： 王帅 | | 班级： 计软18.4 |
| 实验题目：谢林模型模拟 | | | |
| 实验学时：4 | | 实验日期： 2020.3.30 | |
| 实验目的：  模拟谢林模型 | | | |
| 硬件环境：  Pc机 | | | |
| 软件环境：  Dev C++ | | | |
| 实验步骤与内容：  输入：n\*n的矩阵，随机布局的两种节点  输出：  1）调节参数后输出相应的结果  2）需要有界面显示  代码如下：  #include<stdlib.h>  #include<stdio.h>  #include<math.h>  #include<time.h>  #include<unistd.h>  #include<Windows.h>  //宏定义  #define N 30 //矩阵大小  #define K 6 //随机更新的指数（10的 K次方）  #define STATUS 3 //方格状态（空，穷人，富人）  #define EMPTY 0  #define POOR 1  #define RICH 2  #define ECOLOR "47" //三种方格的颜色  #define PCOLOR "44"  #define RCOLOR "41"  #define CLEAR "cls" //Windows清屏命令  int Stage[N][N] = {};  void Initialized\_random(int poor,int rich); //给定穷人和富人的数量，随机初始化stage  void Print\_Stage(); //根据设置的颜色打印stage  double Calc\_thres(int x,int y); //计算（x，y）点的满意度  void Move(int x,int y); //若对当前位置不满意，随机移动到空方格  void Reacter(int x,int y,double thres); //方格中穷人/富人反应器，对当前满意度作出反应 ，选择留下或者随机移动  int Update(char model,double thres); //选择更新模式（随机取点/顺序遍历）和阈值，进行更新    void Initialized\_random(int poor,int rich)  {  int i,j,temp\_status;  for(i=0;i<N;i++)  {  for(j=0;j<N;j++)  {  temp\_status = rand() % STATUS;  while(1)  {  if(temp\_status == 1 && poor > 0)  {  Stage[i][j] = temp\_status;  poor--;  break;  }  else if(temp\_status == 2 && rich > 0)  {  Stage[i][j] = temp\_status;  rich--;  break;  }  else if(temp\_status == 0)  {  Stage[i][j] = temp\_status;  break;  }  else if(poor == 0 || rich == 0) break;  }  }  }  }  void Print\_Stage()  {  int i,j;  for(i=0;i<N;i++)  {  for(j=0;j<N;j++)  {  if(Stage[i][j]==EMPTY) printf("\033[%s;37m \033[0m",ECOLOR);  if(Stage[i][j]==POOR) printf("\033[%s;37m \033[0m",PCOLOR);  if(Stage[i][j]==RICH) printf("\033[%s;37m \033[0m",RCOLOR);  }  printf("\n");  }  }  double Calc\_thres(int x,int y)  {  int Neighbor=0, Same=0;    //左  if(x-1>=0)  {  if(Stage[x-1][y]!=EMPTY)  {  Neighbor++;  if(Stage[x-1][y]==Stage[x][y]) Same++;  }  }    //右  if(x+1<N)  {  if(Stage[x+1][y]!=EMPTY)  {  Neighbor++;  if(Stage[x+1][y]==Stage[x][y]) Same++;  }  }    //下  if(y-1>=0)  {  if(Stage[x][y-1]!=EMPTY)  {  Neighbor++;  if(Stage[x][y-1]==Stage[x][y]) Same++;  }  }    //上  if(y+1<N)  {  if(Stage[x][y+1]!=EMPTY)  {  Neighbor++;  if(Stage[x][y+1]==Stage[x][y]) Same++;  }  }    //左上  if(x-1>=0 && y+1>=0)  {  if(Stage[x-1][y+1]!=EMPTY)  {  Neighbor++;  if(Stage[x-1][y+1]==Stage[x][y]) Same++;  }  }    //右上  if(x+1<N && y+1>=0)  {  if(Stage[x+1][y+1]!=EMPTY)  {  Neighbor++;  if(Stage[x+1][y+1]==Stage[x][y]) Same++;  }  }    //左下  if(x-1>=0 && y-1<N)  {  if(Stage[x-1][y-1]!=EMPTY)  {  Neighbor++;  if(Stage[x-1][y-1]==Stage[x][y]) Same++;  }  }  //右下  if(x+1<N && y-1<N)  {  if(Stage[x+1][y-1]!=EMPTY)  {  Neighbor++;  if(Stage[x+1][y-1]==Stage[x][y]) Same++;  }  }    return (double)Same / Neighbor;  }  void Move(int x,int y)  {  int xn,yn;  while(1)  {  xn = rand()%N;  yn = rand()%N;  if(Stage[xn][yn]==EMPTY)  {  Stage[xn][yn] = Stage[x][y];  Stage[x][y] = EMPTY;  break;  }  }  }  void Reactor(int x,int y,double thres)  {  double th = Calc\_thres(x,y);  if(th<thres) Move(x,y);  }  int Update(char model,double thres)  {  int updated = 0;  if(model == '0') //随机取点  {  while(1)  {  int x = rand()%N,y = rand()%N;  if(Stage[x][y]!=EMPTY)  {  Reactor(x,y,thres);  updated++;  break;  }  }  }  else if(model == '1') //顺序遍历  {  int x,y;  for(x=0;x<N;x++)  {  for(y=0;y<N;y++)  {  if(Stage[x][y]!=EMPTY)  {  Reactor(x,y,thres);  updated++;  }  }  }  }  return updated;  }  int main()  {  int i,j;  int poor,rich,updated;  double thres;  srand((unsigned)time(NULL));    printf("谢林模型矩阵：30\*30\n");    printf("请输入贫穷人数(蓝色)：");  scanf("%d",&poor);  if(poor<=0)  {  printf("输入无效");  return EXIT\_FAILURE;  }    printf("请输入富有人数(红色)：");  scanf("%d",&rich);  if(rich<=0)  {  printf("输入无效");  return EXIT\_FAILURE;  }    printf("请输入搬家的满意度（当同类人/邻居的比例低于此值时就会搬家）：");  scanf("%lf",&thres);  if(thres>1.0||thres<0.0)  {  printf("输入无效");  return EXIT\_FAILURE;  }    Initialized\_random(poor,rich);    for(i=0;i<pow(10,K);i++)  {  updated = Update('0',thres);  if(i == 0 || i%100 == 0)  {  system("cls");  Print\_Stage();  if(i == 0) sleep(3);  else sleep(1);  }  }  }  运行结果：    输入指定参数，会生成一个图像根据两类人的比例动态变化，这里由于视频不好截取，只截了图片，图片如下，会一直变化。 | | | |
| 结论分析与体会：  本次实验让我上课所学到的知识与熟悉的软件编程结合起来，即加强了我的编程能力，又加深了我对谢林模型的理解，更让我感受到了众智科学与网络化产业这门课程的实用性。 | | | |